Domain-Hallucinated Updating for Multi-Domain Face Anti-spoofing

Chengyang Hu[1*†], Ke-Yue Zhang[2†], Taiping Yao[2], Shice Liu[2], Shouhong Ding[2‡], Xin Tan[3], Lizhuang Ma[1,3,4‡]

[1]Shanghai Jiao Tong University
[2]Youtu Lab, Tencent
[3]East China Normal University
[4]MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University
huchengyang@sjtu.edu.cn, {zkyezhang, taipingyao, shiceliu, ericshding}@tencent.com,
xtan@cs.ecnu.edu.cn, ma-lz@cs.sjtu.edu.cn

Abstract

Multi-Domain Face Anti-Spoofing (MD-FAS) is a practical setting that aims to update models on new domains using only novel data while ensuring that the knowledge acquired from previous domains is not forgotten. Prior methods utilize the responses from models to represent the previous domain knowledge or map the different domains into separated feature spaces to prevent forgetting. However, due to domain gaps, the responses of new data are not as accurate as those of previous data. Also, without the supervision of previous data, separated feature spaces might be destroyed by new domains while updating, leading to catastrophic forgetting. Inspired by the challenges posed by the lack of previous data, we solve this issue from a new standpoint that generates hallucinated previous data for updating FAS model. To this end, we propose a novel Domain-Hallucinated Updating (DHU) framework to facilitate the hallucination of data. Specifically, Domain Information Explorer learns representative domain information of the previous domains. Then, Domain Information Hallucination module transfers the new domain data to pseudo-previous domain ones. Moreover, Hallucinated Features Joint Learning module is proposed to asymmetrically align the new and pseudo-previous data for real samples via dual levels to learn more generalized features, promoting the results on all domains. Our experimental results and visualizations demonstrate that the proposed method outperforms state-of-the-art competitors in terms of effectiveness.

Introduction

Face anti-spoofing (FAS) is becoming increasingly important in preventing presentation attacks (PA) on face recognition (FR) technology such as photo, and video replay. To address this issue, researchers distinguish between real people and presentation attacks via deep learning-based methods (Feng et al. 2016; Li et al. 2016; Yang et al. 2014). Nevertheless, they tend to experience performance degradation in more complex real-world scenarios due to domain shifts. To promote performance in the new environment,

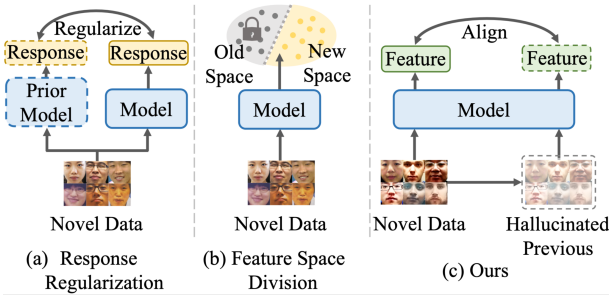


Figure 1: (a) Response regularization-based methods utilize responses of new data on the prior model. (b) Some methods map varied domains' features into different sub-space. (c) Our method hallucinates the previous domain feature and aligns the feature from pseudo-previous and new domain.

researchers have incorporated domain generalization (DG) techniques into FAS to learn features that are applicable across different domains. However, DG FAS (Jia et al. 2020; Zhou et al. 2022b, 2023; Liu et al. 2021a) approaches only utilize the seen data during the training stage, which means they do not effectively utilize the information from novel data. As a result, they often exhibit unsatisfactory performance when applied to new domains. Therefore, in order to improve the performance of FAS on novel domains in real-world scenarios, it is necessary to update the models using collected novel data. The common approach is to retrain the model using both old and new data, such as domain adaptation (DA) methods (Jia et al. 2021; Wang et al. 2021; Zhou et al. 2022a). However, concerns about data privacy policies, particularly for Personally Identifiable Information (PII) like facial images, may prevent access to data from the previous domain during model updating, which might fail DA methods. On the other hand, solely updating models with new data may result in overfitting, causing the model to forget the knowledge acquired from the previous data. Consequently, this might lead to poor generalization performance, which is important for practical FAS applications. Therefore, the preservation of FAS knowledge while updating solely with new data poses significant challenges in model updating, commonly called the plasticity-stability dilemma. This as-

*Work done during an internship at Youtu Lab, Tencent.
†Equal Contributions.
‡Corresponding Authors.

pect is crucial for the practical deployment of FAS and has been explored in initial studies (Guo et al. 2022), referred to as Multi-Domain Face Anti-spoofing (MD-FAS).

To address this issue, researchers have proposed two categories of methods: response regularization-based methods and feature-space-divided methods. Response regularization-based methods utilize responses from the models, such as logits (Li, Hoiem et al. 2016; Dhar et al. 2019), grad-CAM (Selvaraju et al. 2017; Aljundi et al. 2018), or estimated spoof cues (Guo et al. 2022), to maintain performance on the previous domain, as Figure 1 (a). However, due to domain gaps, the responses of new data from the prior model are not as accurate as those of the previous data, which might inhibit learning effective features from the new domain (Aljundi, Chakravarty, and Tuytelaars 2017). While feature-space-divided methods, as Figure 1 (b), leverage isolated parameters (Rebuffi, Bilen, and Vedaldi 2017, 2018; Rusu et al. 2016; Mallya and Lazebnik 2018) or prompts (Wang, Huang, and Hong 2022; Xie, Yan, and He 2022) to continuously map different domains into separate space for updating to novel domains. Nevertheless, without the supervision of previous data, shared parameters in the model may overfit the new data. Consequently, this might compromise the preservation of knowledge of the previous domain (Kanakis et al. 2020). In general, previous methods primarily focus on model improvements to address the challenges. However, none of them adequately compensate for the impact of missing previous data, leading to unsatisfactory results in the previous or new domains.

Considering the unavailability of previous data in MD-FAS, we attempt to address these issues from a novel standpoint. Our approach involves generating hallucinated features of the previous domain during FAS model updating to alleviate the plasticity-stability dilemma more efficiently, as depicted in Figure 1 (c). To this end, we propose a novel Domain-Hallucinated Updating (DHU) framework to facilitate the generation of the hallucinated features. Specifically, to create the pseudo-previous features, Domain Information Explorer is first designed to learn representative domain information from the previous domain for live and spoof data separately. During the updating stage in the new domain, we put forth Domain Information Hallucination module to transfer the features from the new domain to hallucinated features of the previous domain, utilizing the stored domain information. Additionally, in conjunction with asymmetrical supervision on real data, Hallucinated Features Joint Learning module aligns the features of both the new and pseudo-previous domains for real samples at dual levels to learn more generalized features, promoting the results on all domains. Extensive experiments and analysis demonstrate the superiority of our method over the state of the competitors.

The main contributions are summarized as follows:

- We tackle the MD-FAS issue from a new standpoint that generates hallucinated features, alleviating the plasticity-stability dilemma in a more efficient manner.
- We propose a novel framework Domain-Hallucinated Updating (DHU) to learn previous information for live and spoof data separately and then transfer the features from the new domain to hallucinated previous ones. Furthermore, we asymmetrically align the real features of new and pseudo-previous domains to learn more generalized features, promoting the results on all domains.
- Our method achieves promising performance on the FASMD benchmark and extensive experiments demonstrate the effectiveness of our approach.

Related Work

Face Anti-spoofing

Face Anti-spoofing task aims to distinguish between real people and presentation attacks. Previous studies exploit deep-learning-based features (Li et al. 2016; Yang et al. 2014) to capture the spoof cues. Then several auxiliary tasks are introduced to enhance the performance (Zhang et al. 2021a,b), *e.g.* depth map, reflection map, and rPPG signal. Some methods devise novel operators for extracting effective information like CDCN (Yu et al. 2020b) and BCN (Yu et al. 2020a). However, these methods failed in scenarios with domain shifts. To promote the performance on new domains, Domain Generalization (DG) based methods (Chen et al. 2021; Liu et al. 2021b; Wang et al. 2022a) and Domain Adaption (DA) methods (Wang et al. 2021; Li et al. 2018) are proposed. However, DG methods are unable to handle all unseen domains, resulting in subpar performance. DA methods necessitate source data during updating, but accessing this data is not always feasible due to concerns surrounding data privacy policies, which might fail DA methods. Such practical issue has been studied in initial work (Guo et al. 2022), which is called Multi-Domain Learning Face Anti-spoofing (MD-FAS) However, the estimated spoof cues in (Guo et al. 2022) of the new data are still not efficient to prevent forgetting due to the domain gap, which might inhibit learning effective features from the new domain.

Anti-forgetting Learning

Anti-forgetting learning updates models only with novel data to enhance the performance on the new domain while preserving the knowledge acquired from previous domains, which alleviates the plasticity-stability dilemma. Response regularization-based methods utilize responses from the prior model, such as logits (Li, Hoiem et al. 2016; Dhar et al. 2019), grad-CAM (Selvaraju et al. 2017; Aljundi et al. 2018), and estimated spoof cues (Guo et al. 2022), to prevent forgetting. However, due to the domain gap, the responses of new data may not be as accurate as those of previous data, which might inhibit learning effective features from the novel domain. Feature-space-divided methods separate the feature space for different domains to prevent forgetting via isolated parameters (Rebuffi, Bilen, and Vedaldi 2017, 2018; Rusu et al. 2016; Mallya and Lazebnik 2018) and prompts (Wang, Huang, and Hong 2022; Xie, Yan, and He 2022). But, without the supervision of previous data, shared parameters might overfit new domain data, which inevitably causes catastrophic forgetting. Inspired by the impact of previous data, we alleviate this dilemma from a novel standpoint that hallucinates the previous domain feature on new domains to prohibit forgetting in a more efficient way.

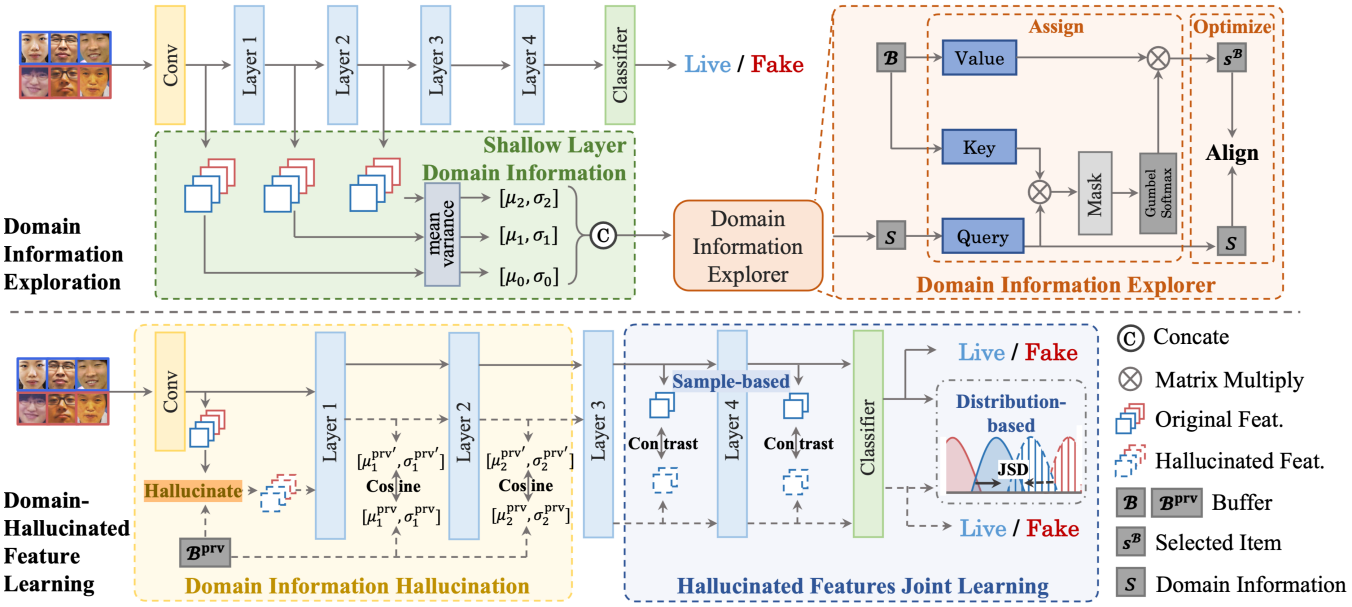


Figure 2: The overall structure of proposed Domain-Hallucinated Updating (DHU) framework. First, we propose Domain Information Explorer to learn effective domain information in shallow layers. Also, we first propose Domain Information Hallucination module to hallucinate the previous domain feature to prevent the forgetting issue. After obtaining the pseudo-previous and new domain feature, we design Hallucinated Features Joint Learning to align the real input features from sample-based view and distribution-based view for generalizability.

Method

In this section, we introduce the proposed Domain-Hallucinated Updating (DHU) framework in detail. As depicted in Figure 2, we first devise Domain Information Explorer to learn representative domain information from the current training domain. Then, Domain Information Hallucination module is proposed to generate the pseudo-previous features via stored domain information and utilize the hallucinated feature to tackle the catastrophic-forgetting issue. Furthermore, to take advantage of novel and hallucinate previous features, Hallucinated Features Joint Learning module is presented to align features in an asymmetrical manner from the sample-based view and distribution-based view to improve the generalizability.

Domain Information Exploration

Definition of Domain Information. Given the input $x \in \mathbb{R}^{c\times h\times w}$, we utilize a feature extractor $F(x, \theta_F)$ to extract the multi-layer feature as $\{f_i \in \mathbb{R}^{c_i\times h_i\times w_i} \mid i \in [0, l]\}$, where f_i is the output feature from i-th layer. c_i, h_i, w_i are the feature's channel number, width, and height. Since shallow layers tend to contain more domain-specific information compared to deep layers, we focus on features from the first l layers to collect relevant domain information. Following the work (Wang et al. 2022b), we define the domain information of a feature based on its mean and variance.

$$\mu_i = \frac{1}{h_i \cdot w_i}\sum_{h_i}\sum_{w_i} f_i, \quad \sigma_i^2 = \frac{1}{h_i \cdot w_i}\sum_{h_i}\sum_{w_i}(f_i - \mu_i)^2, \tag{1}$$

where $\mu_i \in \mathbb{R}^{c_i}, \sigma_i \in \mathbb{R}^{c_i}$. The domain information of the feature is defined as the concatenation of its mean and variance from all shallow layers:

$$s = \text{concat}([\mu_0, \sigma_0, \cdots, \mu_l, \sigma_l]) \in \mathbb{R}^{d_s}, \tag{2}$$

where $\text{concat}(\cdot)$ is the concatenation operation, $d_s = 2\sum_i c_i$ is the dimension of the domain information.

Domain Information Explorer. To obtain the representative domain information from training data, we propose Domain Information Explorer to capture such information. First, we define a learnable buffer as $\mathcal{B} = \{\mathcal{B}_i \in \mathbb{R}^{N_{\mathcal{B}_i}\times d_s} \mid i \in [0, 1]\}$, to extract and store domain information, where i is the task-related label (0 as real, 1 as fake), $N_{\mathcal{B}} = N_{\mathcal{B}_0} + N_{\mathcal{B}_1}$ is the buffer size. Then we simultaneously optimize the model θ_F and buffer $\mathcal{B}$ via two steps: **assigning**, and **optimizing**. Once the training is completed, the learned buffer $\mathcal{B}$ already captures the current domain information. Finally, we introduce the merging step to combine the learned buffer $\mathcal{B}$ with the previous domain buffer $\mathcal{B}^{\text{prv}}$, resulting in more comprehensive domain information buffer.

Assigning step. First, we assign the domain information of each sample, extracted by θ'_F, to a corresponding item $s_k^{\mathcal{B}}$ in buffer $\mathcal{B}$. The assign function is:

$$\begin{aligned} s &= \arg\max_k P(s_k^{\mathcal{B}} \mid s_x, y; \theta'_F) \\ &= \arg\max_k \frac{P(s_x \mid s_k^{\mathcal{B}}, y; \theta'_F)P(s_k^{\mathcal{B}} \mid y)}{\sum_{i=1}^{N_{\mathcal{B}}} P(s_x \mid s_i^{\mathcal{B}}, y; \theta'_F)P(s_i^{\mathcal{B}} \mid y)} \\ &= \arg\max_k \hat{s}_x^\top \hat{s}_k^{\mathcal{B}_y}, \end{aligned} \tag{3}$$

where s_x is calculated via Eq. 1, 2, $s_k^{\mathcal{B}}$ is the k-th item in buffer $\mathcal{B}$. $\hat{s} = \frac{s}{\|s\|_2}$ is the normalized domain information.

The assigned item $s_k^{\mathcal{B}}$ has the highest similarity with the domain information of the sample. To implement the assign function as a forwarding procedure, we utilize Gumbel-Softmax (Jang, Gu, and Poole 2016) to ensure that Domain Information Explorer selects only one item $s_k^{\mathcal{B}}$ from the buffer $\mathcal{B}$ while still allowing gradient backpropagation. The procedure is formulated as:

$$\begin{aligned}\hat{s} &= \text{GumbelSoftmax}(\hat{s}_x^\top \hat{s}^{\mathcal{B}_y}) \cdot \hat{s}^{\mathcal{B}_y} \\ &= \text{GumbelSoftmax}(\text{Mask}(y) \cdot \hat{s}_x^\top \hat{s}^{\mathcal{B}}) \cdot \hat{s}^{\mathcal{B}} \\ &= \text{GumbelSoftmax}(\text{Mask}(y) \cdot \mathbf{Q}^\top \mathbf{K}) \cdot \mathbf{V},\end{aligned} \tag{4}$$

$\text{Mask}(y)$ is a boolean function to mask off stored domain information with different labels. Finally, the assign function is formulated using the attention module, where the query is $\hat{s}_x$, and the key and value are $\hat{s}^{\mathcal{B}}$.

Optimizing step. Based on the assignment of each sample, we optimize the buffer $\mathcal{B}$. First, to ensure the buffer $\mathcal{B}$ could represent the overall domain information of training data, we optimize each assigned item $s_k^{\mathcal{B}}$ in buffer $\mathcal{B}$ to ensure the selected item is representative to corresponding sample:

$$\begin{aligned}\mathcal{L}_{\text{d-info}} &= \mathbb{E}_{(x,y)\sim\mathbb{D}} P(s_k^{\mathcal{B}} | s_x, y; \theta'_F) \\ &= \frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{\kappa \cdot \hat{s}_{x_i}^\top \hat{s}_k^{\mathcal{B}}}}{\sum_{j=1}^{N_{\mathcal{B}}} e^{\kappa \cdot \hat{s}_{x_i}^\top \hat{s}_j^{\mathcal{B}}}},\end{aligned} \tag{5}$$

where N is the batch size, κ is the temperature.

However, the learned buffer $\mathcal{B}$ may drop into a trivial solution. For instance, only one item $s_k^{\mathcal{B}}$ in the buffer $\mathcal{B}$ is selected and assigned with all input samples. To prevent trivial solutions, we utilize entropy loss to assign the samples to the buffer evenly:

$$\mathcal{L}_{\text{entropy}} = \frac{1}{N_{\mathcal{B}}}\sum_{i=1}^{N_{\mathcal{B}}} (\hat{s}^\top \cdot \hat{s}_i^{\mathcal{B}}) \log(\hat{s}^\top \cdot \hat{s}_i^{\mathcal{B}}). \tag{6}$$

Moreover, to enhance the initialization of buffer $\mathcal{B}$ for capturing more comprehensive domain information, we utilize the domain information from all samples and apply the K-means (Hartigan and Wong 1979; Arthur and Vassilvitskii 2007) algorithm to obtain the centroid, which serves as the initial value of the buffer $\mathcal{B}$.

Merging step. Once updating to a novel domain, we get one new buffer $\mathcal{B}$ of novel data. It's impractical to save all the domains buffer $\mathcal{B}$ since limited space during deployment. Hence we apply K-means algorithm to select representative buffers from the previous buffer $\mathcal{B}^{\text{prv}}$ and new buffer $\mathcal{B}$.

Algorithm 1: Training Procedure of DHU

Input: Datasets $\{\mathbb{D}^1, \mathbb{D}^2, \cdots, \mathbb{D}^n\}$

Initialize the buffer $\mathcal{B}^{\text{prv}}$;
for $i = 1 : n$ **do**
  Initialize the buffer $\mathcal{B}$;
  while *Training Steps* **do**
    `// Assign step:`
    Compute s of $x \sim \mathbb{D}^i$ with extractor θ_F by Eq. 1, 2;
    Assign s_i to one item in buffer $\mathcal{B}$ via Eq. 3;
    `// Optimize Domain Information Explorer:`
    Compute the total loss to optimize $\mathcal{B}$: $\mathcal{L}_{\mathcal{B}} = \mathcal{L}_{\text{d-info}} + \mathcal{L}_{\text{entropy}}$;
    $\mathcal{B}' \leftarrow \mathcal{B} - \nabla_{\mathcal{B}} \mathcal{L}_{\mathcal{B}}$;
    `// Hallucinated Features Joint Learning:`
    Select one item in buffer $\mathcal{B}^{\text{prv}}$ for $x \sim \mathbb{D}^i$ via Eq. 7;
    Generate pseudo-previous features via Eq. 8;
    Compute the total loss: $\mathcal{L}_{\theta'_F} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{hal}} + \mathcal{L}_{\text{smp}} + \mathcal{L}_{\text{dist}}$;
    $\theta'_F \leftarrow \theta_F - \nabla_{\theta_F} \mathcal{L}_{\theta_F}$;
  end
  `// Merging Domain Information:`
  $\mathcal{B}^{\text{prv}} \leftarrow \text{Kmeans}(\mathcal{B}^{\text{prv}}, \mathcal{B})$
end

Domain-Hallucinated Feature Learning

Domain Information Hallucination. Domain Information Hallucination module generates the pseudo-previous feature (denoted as *prv*) by combining new features (denoted as *new*) with stored buffer $\mathcal{B}^{\text{prv}}$. Specifically, to synthesize the previous-domain feature efficiently, we select a style item $s_k^{\mathcal{B}_y^{\text{prv}}}$ from the buffer $\mathcal{B}^{\text{prv}}$ according to the similarity between the domain information of new samples and buffer. The higher the similarity, the easier the style conversion becomes, and the less information is lost.

$$s^{\text{prv}} = \arg\max_k \cos(s^{\text{new}}, s_k^{\mathcal{B}_y^{\text{prv}}}), \tag{7}$$

where s^{new} are the domain information of novel data, s^{prv} are the selected item from the buffer $\mathcal{B}^{\text{prv}}$, $\cos(\cdot)$ is to calculate cosine similarity. After obtaining the most similar previous item s^{prv}, we generate the feature in the initial layer:

$$f_0^{\text{prv'}} = \frac{f_0^{\text{new}} - \mu_0^{\text{new}}}{\sigma_0^{\text{new}}} \cdot \lambda_\sigma \cdot \sigma_0^{\text{prv}} + \lambda_\mu \cdot \mu_0^{\text{prv}}, \tag{8}$$

where $f_0^{\text{prv'}}$ is the simulated previous domain feature, $\lambda_\sigma = |\sigma_0^{\text{new}}|/|\sigma_0^{\text{prv}}|$, $\lambda_\mu = |\mu_0^{\text{new}}|/|\mu_0^{\text{prv}}|$. Since we select items based on normalized features, it is necessary to rescale the selected item to match the scale of the new feature. Then the hallucinated feature of the previous domain $f_0^{\text{prv'}}$ is fed into the extractor θ_F, getting features $\{f_i^{\text{prv'}} | i \in [1, l]\}$. For each $f_i^{\text{prv'}}$, we calculate domain information $s^{\text{prv'}}$ via Eq. 1,

Method	Spoof (A→B)			Ethnicity (A→C)			Age (A→D)			Illumination (A→E)			Average		
	A	B	avg	A	C	avg	A	D	avg	A	E	avg	A	B-E	avg
Joint	94.0	86.9	90.5	94.1	71.8	83.0	94.1	69.5	81.8	97.6	95.5	96.1	95.0	80.9	88.0
Source	97.4	-	-	97.4	-	-	97.4	-	-	97.4	-	-	97.0	-	-
Finetune	74.2	82.1	78.2	79.2	73.9	76.6	86.6	67.6	77.1	88.4	89.0	88.7	82.1	78.2	80.1
LwF (2016)	86.6	77.6	82.1	86.2	73.6	79.9	89.3	57.1	73.2	89.1	90.0	89.6	87.8	74.5	81.2
LwM (2019)	94.4	72.4	83.4	92.3	66.4	79.3	90.3	59.3	74.8	94.9	89.6	92.3	93.1	71.8	82.5
MAS (2018)	90.1	82.1	86.1	82.5	74.7	78.6	91.8	67.8	79.8	92.0	83.7	87.9	89.1	77.1	83.1
FAS-wrapper† (2022)	89.7	70.7	80.2	91.2	73.9	82.6	92.2	66.2	79.2	93.2	90.7	92.0	91.6	75.4	83.5
ER (2019)	92.8	80.0	86.4	91.9	71.4	81.7	90.2	67.6	78.9	90.9	90.3	90.6	91.5	77.3	84.4
DER (2020)	88.4	70.0	79.2	93.4	61.8	77.6	90.1	61.5	75.8	94.4	85.3	89.8	91.5	69.6	80.6
GEM (2017)	95.3	75.2	85.3	90.7	71.8	81.2	91.9	66.8	79.3	95.1	89.9	92.6	93.3	75.9	84.6
A-GEM (2019)	**95.5**	76.5	86.0	92.9	73.0	83.0	92.3	66.3	79.3	94.4	90.4	92.4	93.8	76.6	85.1
Seri. Adapter (2017)	86.8	77.6	82.2	89.6	70.6	80.1	89.8	67.2	78.5	90.5	88.5	89.5	89.2	77.2	83.2
Para Adapter (2018)	88.5	82.6	85.6	90.3	70.6	80.5	90.2	68.3	79.3	90.4	84.0	87.2	89.9	76.4	83.1
Ours	90.2	**83.1**	**86.7**	**96.2**	**74.7**	**85.5**	**96.9**	**68.9**	**82.9**	**95.2**	**92.6**	**93.9**	**94.6**	**79.8**	**87.2**

Table 1: The performance reported in TPR@FPR=0.5%. The model first trains on the source dataset (A) and then train on other datasets (B-E). avg indicates the average performance on two datasets. Bolded scores indicate the best performance. †: We re-implement FAS-wrapper with ResNet-18 by inserting stacked CNN models as Discriminator into every ResNet layer.

2. Then, we encourage the shadow layers of extractor θ_F extract the features containing similar domain information $s^{\text{prv'}}$ with assigned item s^{prv} to preserve knowledge from previous domains, which is formulated as:

$$\mathcal{L}_{\text{hal}} = 1 - \cos(s^{\text{prv'}}, s^{\text{prv}}). \tag{9}$$

Hallucinated Features Joint Learning. Considering that utilizing pseudo previous feature jointly learning with the new domain might overfit these data and hinder the generalizability, we propose Hallucinated Features Joint Learning module to tackle this issue. With the concern of larger domain distribution discrepancies in presentation attacks (Jia et al. 2020), we only align the features of real samples in an asymmetrical manner. On the one hand, the pseudo-previous feature and corresponding novel feature should be close in the feature space of deep layers. On the other hand, the whole distribution from different domains should be consistent. Therefore we align the real features from dual views including sample-based and distribution-based views. First, we align the real sample features in deep layers (*i.e.* from the l-th layer to the last layer) that contains more task-related information in sample-based view via contrastive learning:

$$\mathcal{L}_{\text{smp}} = -\frac{1}{N}\sum_{y_i=0}\log\frac{e^{\kappa\cdot(f_{x_i}^{\text{prv'}})^\top f_{x_i}^{\text{new}}}}{\sum_{y_j=0} e^{\kappa\cdot(f_{x_i}^{\text{prv'}})^\top f_{x_j}^{\text{new}}} + e^{\kappa\cdot(f_{x_j}^{\text{prv'}})^\top f_{x_i}^{\text{new}}}}, \tag{10}$$

where $f_{x_i}^{\text{new}}, f_{x_i}^{\text{prv'}}$ are the l-th to last layer features from input x_i of new domain and pesudo-previous feature, κ is the temperature. Then, to ensure consistency in the overall distribution of the real input, we utilize Jensen-Shannon Divergence to constrain logits from different domains from a distribution-based view:

$$\mathcal{L}_{\text{dist}} = \text{JSD}_{y_i=0}(z_i^{\text{prv'}} \| z_i^{\text{new}}), \tag{11}$$

where $\text{JSD}(\cdot||\cdot)$ is Jensen-Shannon Divergence, $z_i^{\text{prv'}}$ and z_i^{new} are the logits from the real input features. Also, the pseudo-previous features should have the same prediction as the new domain feature:

$$\mathcal{L}_{\text{cls}} = -\,y_i\log(\text{BC}(z_i^{\text{new}})) - y_i\log(\text{BC}(z_i^{\text{prv'}})). \tag{12}$$

Method	AUC↑		ACER↓	
	Mean	Forget	Mean	Forget
Joint	86.8	-	17.2	-
Finetune	83.0	15.7	21.0	16.1
LwF (2016)	84.2	15.3	21.6	14.6
LwM (2019)	84.1	15.1	21.2	10.1
MAS (2018)	83.2	13.9	22.4	11.7
ER (2019)	84.0	**5.3**	22.1	**6.3**
DER (2020)	81.4	11.2	22.2	10.4
GEM (2017)	81.3	15.0	24.4	12.4
A-GEM (2019)	83.5	11.3	21.2	12.6
Ours	**84.6**	<u>10.9</u>	**20.8**	<u>9.6</u>

Table 2: The performance reported in AUC and ACER on long sequence CASIA→OULU→Idiap→MSU. Bolded scores show the best performance. Underlined scores show the second best performance.

Training Procedure

The overall training procedure is shown in Algorithm 1. In each training stage, we learn the representative domain information buffer $\mathcal{B}$ via Domain Information Explorer and train the model via Domain-Hallucinated Feature Learning.

Experiments

Experimental Setup

Databases. Following the previous work (Guo et al. 2022), we utilize the FASMD dataset based on SiW (Liu, Jourabloo, and Liu 2018), SiW-Mv2 (Liu et al. 2019) and OULU-NPU (Boulkenafet et al. 2017) to evaluate the proposed methods. The dataset is divided into five parts, including the source dataset (A), dataset with new spoof type (B), with new ethnicity distribution (C), with new age distribution (D), and with new illumination (E). Strictly following the setting (Guo et al. 2022), we begin by training on the source dataset (A) and subsequently on other datasets with varying domain distributions (B-E). There are four protocols in place: namely A→B, A→C, A→D, A→E. Besides,

Method	TPR↑				ACER↓			
	C	D	E	avg	C	D	E	avg
Joint	19.9	46.9	54.6	40.5	18.6	20.6	11.1	16.8
Source	65.6	18.2	91.4	58.4	6.8	10.5	3.0	6.8
Finetune	44.8	31.2	91.4	55.8	18.2	10.3	3.3	10.6
LwF	80.2	37.1	95.2	70.8	6.7	7.5	2.8	5.7
LwM	79.8	42.9	94.7	72.5	**6.6**	6.6	2.8	**5.3**
MAS	41.5	43.9	75.8	53.7	13.7	10.6	9.0	11.1
FAS-warpper	59.3	45.6	85.2	63.5	8.3	6.2	4.2	6.2
ER	51.9	68.8	96.0	82.4	8.4	6.6	2.5	5.8
DER	58.9	37.4	72.4	56.2	8.8	16.2	6.1	10.4
GEM	58.1	47.0	83.9	66.3	9.3	7.6	3.5	6.8
A-GEM	66.8	37.9	85.3	63.3	10.9	10.8	4.2	8.6
Seri. Adapter	52.3	35.7	87.2	58.4	9.6	10.7	4.6	8.3
Para Adapter	56.1	33.2	86.6	58.6	9.1	10.9	4.3	8.1
Ours	**80.5**	**70.9**	**97.5**	**83.0**	8.5	**6.0**	**1.4**	**5.3**

Table 3: The performance on unseen domains in TPR@FPR=0.5% and ACER. The model first trains sequentially with datasets A and B and tests on C-E. The result in Serial, and Parallel Res-Adapter is the average result with different adapters.

$\mathcal{L}_{cls}$	$\mathcal{L}_{hal}$	$\mathcal{L}_{smp}$	$\mathcal{L}_{dist}$	A	C	avg
✓				90.2	67.6	78.9
✓	✓			93.0	71.8	82.4
✓	✓	✓		95.2	73.4	84.3
✓	✓		✓	94.8	73.9	84.4
✓		✓		92.2	71.3	81.8
✓			✓	91.0	71.7	81.4
✓		✓	✓	92.6	71.2	81.9
✓	✓	✓	✓	**96.2**	**74.7**	**85.5**

Table 4: Ablation study on loss functions under A→C.

Learning Strategy	A	C	avg
Random	68.3	73.9	71.1
Kmeans (1979)	82.7	75.1	78.9
DeepCluster (2018)	89.5	74.3	81.9
DHU(Ours)	**93.1**	**75.5**	**84.3**

Table 5: Ablation Study on different methods to obtain domain information in the previous training stage.

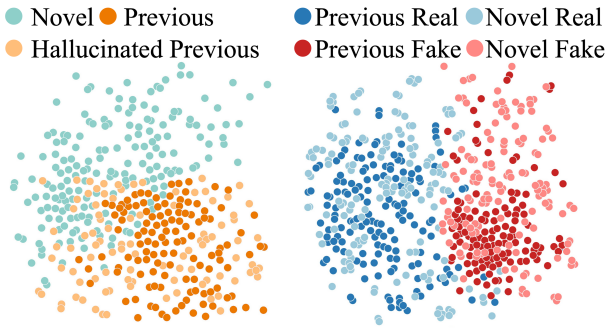
to evaluate the adaptation abilities of models in the context of continuous domain changing, we conduct the experiment on long sequences using four datasets including OULU-NPU (Boulkenafet et al. 2017), CASIA-FASD (Zhang et al. 2012), Idiap Replay-Attack (Chingovska, Anjos, and Marcel 2012) and MSU-MFSD (Wen et al. 2015) in sequence CASIA→OULU→Idiap→MSU. Furthermore, we evaluate the generalization capabilities of methods based on the aforementioned setting. This involves training on datasets A and B sequentially and testing on unseen datasets C-E. We apply Attack Presentation Classification Error Rate (APCER), Bona Fide Presentation Classification Error Rate (BPCER), Average Classification Error Rate (ACER), Area Under Curve (AUC), and True Positive Rate at False Positive Rate = 0.5% (TPR@FPR=0.5%) as metrics.

Implementation Details. The input is detected face region normalized to size 256×256 with RGB channels. The extractor used is ResNet18 (He et al. 2016) with 4 layers. The size of buffer $N_{\mathcal{B}}$ is 200 with $N_{\mathcal{B}_0} : N_{\mathcal{B}_1} = 1 : 1$ and the dimension of domain information is $d_s = 512$ from the first 2 layers. The batch size is 16, and the ratio of the real and fake images is $1 : 1$. Strictly following (Guo et al. 2022), the ratio of the images from OULU-NPU, SiW, and SiW-Mv2 is set to $1 : 1 : 2$. We set $l = 2$, $\kappa = 1/0.7$. The learning rate is 1e-2 and we train each dataset for 50,000 steps. We use the public Pytorch (Paszke et al. 2017) framework with 32G Tesla V100 on Linux OS to implement our framework.

Comparison Results

Performance on Anti-forgetting Setting. We follow the previous work (Guo et al. 2022), and test our methods with the following methods: "Joint" trains on the source and target together; "Source" only trains on the source; "Finetune" first trains on the source and simply updates on the target, which is the upper bounding for simultaneous training on both source and target. The compared methods include response regularization-based methods like LwF (Li, Hoiem et al. 2016), LwM (Dhar et al. 2019) and MAS (Aljundi et al. 2018), FAS-warpper (Guo et al. 2022). Also, we compared with some parameter isolation methods like Serial and Parallel Res-Adapter (Rebuffi, Bilen, and Vedaldi 2017, 2018). Extensively, we compare with replay-based methods which store a small size of source data in the buffer. We select ER (Riemer et al. 2019), DER (Buzzega et al. 2020), GEM (Lopez-Paz and Ranzato 2017), A-GEM (Chaudhry et al. 2019) with the buffer size 200. Table 1 shows the performance of TPR@FPR = 0.5% on four protocols. We have the following observations: 1) Response regularization-based methods show worse performance because the response is untrustworthy due to the domain gap. 2) Replay-based methods have better performance but suffer an overfitting issue for small buffer data. Also, they need to store many exemplars from the previous dataset. 3) Parameter-isolated methods that shares parameters may overfit the new domain which would hinder source domain performance. 4) Our method shows the best result, which indicates that DHU framework has a better ability to tackle plasticity-stability dilemma. Especially, our methods surpass the best previous methods with 0.3%, 2.5%, 3.1%, 1.3%, and 2.1% on average. Also, we provide the performance of long sequence updating in Table 2. Our method shows the best performance compared to previous methods in average performance and demonstrates comparable results with memory-based methods in forgetting. Different from such methods, like ER (Riemer et al. 2019), storing previous images to prevent forgetting, we only store some effective buffers to achieve the comparable effect of preservation on previous knowledge, which verifies the effectiveness of our method.

Performance on Generalization Setting. To verify the generalization ability, we extensively generate three protocols. We first sequentially train the model with datasets A and B, then test on datasets with unseen different domains C-E, with the metrics TPR@FPR=0.5% and ACER. In Table 3,



(a) Shallow Layer Feature f_2 (b) Deep Layer Feature f_4

Figure 3: The t-SNE visualization of the feature from different layers. (a) shows the previous, hallucinated previous, and new feature in the shallow layer (f_2) (b) shows the previous and new feature in the deep layer (f_4).

our method indicates the best performance compared to previous methods. Our method proposed Hallucinated Features Joint Learning module to align real input features from different domains that improve generalizability. However, previous methods do not contain a special design to tackle the generalizable issue. For instance, response regularization-based methods failed to generalize due to the domain gap which cause responses meaningless to the novel domain and obstacles to the model learning generalizable features. Replay-based methods with the limited data size may cause the overfitting issue and degrade model generalizability.

Ablation Study

Study on Different Loss Functions. We verify the effectiveness of different loss functions on A→C task. As shown in Table 4, we have the following observations: 1) Replaying the domain information with $\mathcal{L}_{\text{hal}}$ significantly enhances the preservation of knowledge of the previous domain. 2) $\mathcal{L}_{\text{smp}}$ and $\mathcal{L}_{\text{dist}}$ improve the average performance on two domain datasets which reflects that joint feature alignment indeed improves the performance on both domains. 3) With all loss functions, our method shows the best performance, which verifies that all loss functions benefit the performance and play important roles in the proposed network.

Study on Domain Information Learning Strategy. In this section, we explore the influence of different domain information exploration strategies on A→C setting. In "Random" strategy, we randomly store the domain information of 200 samples. In "Kmeans" strategy, we extract the domain information from all data and run K-means algorithm to obtain the 200 centroids. In "DeepCluster" (Caron et al. 2018) strategy, we run the K-means every epoch to assign the domain information with pseudo label and train a classifier to predict the pseudo label. As shown in Table 5, each method exhibits similar performance on new domain C. However, compared with DHU, all of them experience a significant drop in performance on the previous domain A. Our method learns better buffer $\mathcal{B}$ through Domain Information Explorer, which shows the best performance on the previous domain.

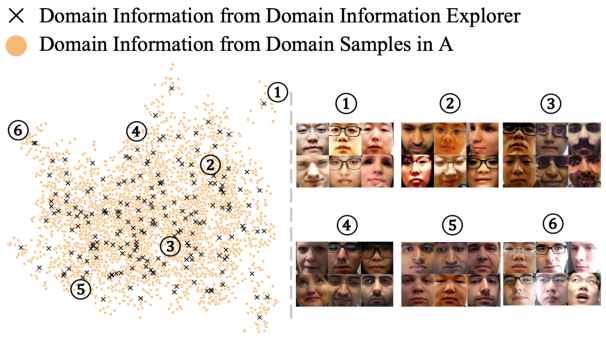


Figure 4: The t-SNE visualization of the previous domain information distribution. The right pictures show examples from different domain information groups.

Visualization and Analysis

Analysis of the Feature Distribution. To verify the effectiveness of Domain Information Hallucination module and Hallucinated Features Joint Learning module, we visualize the feature distribution from different layers via t-SNE (Maaten and Hinton 2008) in Figure 3. (a) shows the inferred shallow features f_2 of the model. Due to the constraints of the $\mathcal{L}_{\text{hal}}$, the network still generates features similar to the original previous ones, which avoids the forgetting problem. (b) shows the deep features f_4. There is a clear boundary between real and fake samples, and the distributions of real samples in different domains are more consistent than presentation attacks, which indicates the generalizability of the Hallucinated Features Joint Learning module.

Visualization of Domain Information. We visualize the learned buffer $\mathcal{B}$ in the feature space with previous domain data to gain insights into domain information that is learned in buffer $\mathcal{B}$. As shown in Figure 4, the learned buffer $\mathcal{B}$ covers the whole domain information distribution of previous domain data. Also, we display some samples assigned to the same buffer, demonstrating a high level of consistency. For example, in Domain Information Group 1, the samples contain a higher brightness, while the samples in Domain Information Group 3 contain a lower brightness.

Conclusion

In this paper, we propose a novel Domain-Hallucinated Updating (DHU) framework to address the challenging task of Multi-domain Face Anti-Spoofing (MD-FAS). First, we introduce Domain Information Explorer in the previous training stage that learns representative domain information buffer $\mathcal{B}$. Then, Domain Information Hallucination module is designed in the new training stage to generate pseudo-previous domain information to prevent forgetting. Additionally, we devise Hallucinated Features Joint Learning module to utilize pseudo-previous and new domain features, which aligns real samples' features from sample-based and distribution-based views to improve model's generalizability. The experimental results and visualizations demonstrate that the proposed method outperforms other competitors.

Acknowledgements

This work is supported by National Natural Science Foundation of China (No. 72192821, No. 62302167), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), Shanghai Science and Technology Commission (21511101200), Shanghai Sailing Program (23YF1410500) and CCF-Tencent Rhino-Bird Young Faculty Open Research Fund (RAGR20230121).

References

Aljundi, R.; Babiloni, F.; Elhoseiny, M.; Rohrbach, M.; and Tuytelaars, T. 2018. Memory Aware Synapses: Learning What (not) to Forget. In *ECCV*.

Aljundi, R.; Chakravarty, P.; and Tuytelaars, T. 2017. Expert Gate: Lifelong Learning with a Network of Experts. In *CVPR*.

Arthur, D.; and Vassilvitskii, S. 2007. K-means++ the advantages of careful seeding. In *ACM-SIAM*.

Boulkenafet, Z.; Komulainen, J.; Li, L.; Feng, X.; and Hadid, A. 2017. Oulu-npu: A mobile face presentation attack database with real-world variations. In *FG*.

Buzzega, P.; Boschini, M.; Porrello, A.; Abati, D.; and Calderara, S. 2020. Dark Experience for General Continual Learning: a Strong, Simple Baseline. In *NIPS*.

Caron, M.; Bojanowski, P.; Joulin, A.; and Douze, M. 2018. Deep Clustering for Unsupervised Learning of Visual Features. In *ECCV*.

Chaudhry, A.; Ranzato, M.; Rohrbach, M.; and Elhoseiny, M. 2019. Efficient Lifelong Learning with A-GEM. In *ICLR*.

Chen, Z.; Yao, T.; Sheng, K.; Ding, S.; Tai, Y.; Li, J.; Huang, F.; and Jin, X. 2021. Generalizable Representation Learning for Mixture Domain Face Anti-Spoofing. In *AAAI*.

Chingovska, I.; Anjos, A.; and Marcel, S. 2012. On the effectiveness of local binary patterns in face anti-spoofing. In *BIOSIG*.

Dhar, P.; Singh, R. V.; Peng, K.; Wu, Z.; and Chellappa, R. 2019. Learning Without Memorizing. In *CVPR*.

Feng, L.; Po, L.-M.; Li, Y.; Xu, X.; Yuan, F.; Cheung, T. C.-H.; and Cheung, K.-W. 2016. Integration of image quality and motion cues for face anti-spoofing: A neural network approach. In *JVCIR*.

Guo, X.; Liu, Y.; Jain, A. K.; and Liu, X. 2022. Multi-domain Learning for Updating Face Anti-spoofing Models. In *ECCV*.

Hartigan, J. A.; and Wong, M. A. 1979. Algorithm AS 136: A k-means clustering algorithm. In *booktitle of the royal statistical society. series c (applied statistics)*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Jang, E.; Gu, S.; and Poole, B. 2016. Categorical reparameterization with gumbel-softmax. In *arXiv*.

Jia, Y.; Zhang, J.; Shan, S.; and Chen, X. 2020. Single-Side Domain Generalization for Face Anti-Spoofing. In *CVPR*.

Jia, Y.; Zhang, J.; Shan, S.; and Chen, X. 2021. Unified unsupervised and semi-supervised domain adaptation network for cross-scenario face anti-spoofing. In *PR*.

Kanakis, M.; Brüggemann, D.; Saha, S.; Georgoulis, S.; Obukhov, A.; and Gool, L. V. 2020. Reparameterizing Convolutions for Incremental Multi-Task Learning Without Task Interference. In *ECCV*.

Li, H.; Li, W.; Cao, H.; Wang, S.; Huang, F.; and Kot, A. C. 2018. Unsupervised Domain Adaptation for Face Anti-Spoofing.

Li, L.; Feng, X.; Boulkenafet, Z.; Xia, Z.; Li, M.; and Hadid, A. 2016. An original face anti-spoofing approach using partial convolutional neural network. In *IPTA*.

Li, Z.; Hoiem, D.; et al. 2016. Learning Without Forgetting. In *ECCV*.

Liu, S.; Zhang, K.-Y.; Yao, T.; Bi, M.; Ding, S.; Li, J.; Huang, F.; and Ma, L. 2021a. Adaptive Normalized Representation Learning for Generalizable Face Anti-Spoofing. In *ACM MM*.

Liu, S.; Zhang, K.-Y.; Yao, T.; Sheng, K.; Ding, S.; Tai, Y.; Li, J.; Xie, Y.; and Ma, L. 2021b. Dual reweighting domain generalization for face presentation attack detection. In *IJCAI*.

Liu, Y.; Jourabloo, A.; and Liu, X. 2018. Learning Deep Models for Face Anti-Spoofing: Binary or Auxiliary Supervision. In *CVPR*.

Liu, Y.; Stehouwer, J.; Jourabloo, A.; and Liu, X. 2019. Deep Tree Learning for Zero-Shot Face Anti-Spoofing. In *CVPR*.

Lopez-Paz, D.; and Ranzato, M. 2017. Gradient Episodic Memory for Continual Learning. In *NIPS*.

Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. In *booktitle of machine learning research*.

Mallya, A.; and Lazebnik, S. 2018. Packnet: Adding multiple tasks to a single network by iterative pruning. In *CVPR*.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch. In *arXiv*.

Rebuffi, S.; Bilen, H.; and Vedaldi, A. 2017. Learning multiple visual domains with residual adapters. In *NIPS*.

Rebuffi, S.; Bilen, H.; and Vedaldi, A. 2018. Efficient Parametrization of Multi-Domain Deep Neural Networks. In *CVPR*.

Riemer, M.; Cases, I.; Ajemian, R.; Liu, M.; Rish, I.; Tu, Y.; and Tesauro, G. 2019. Learning to Learn without Forgetting by Maximizing Transfer and Minimizing Interference. In *ICLR*.

Rusu, A. A.; Rabinowitz, N. C.; Desjardins, G.; Soyer, H.; Kirkpatrick, J.; Kavukcuoglu, K.; Pascanu, R.; and Hadsell, R. 2016. Progressive Neural Networks. In *arXiv*.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *ICCV*.

Wang, C.-Y.; Lu, Y.-D.; Yang, S.-T.; and Lai, S.-H. 2022a. PatchNet: A Simple Face Anti-Spoofing Framework via Fine-Grained Patch Recognition. In *CVPR*.

Wang, G.; Han, H.; Shan, S.; and Chen, X. 2021. Unsupervised Adversarial Domain Adaptation for Cross-Domain Face Presentation Attack Detection. In *TIFS*.

Wang, Y.; Huang, Z.; and Hong, X. 2022. S-Prompts Learning with Pre-trained Transformers: An Occam's Razor for Domain Incremental Learning. In *CoRR*.

Wang, Z.; Wang, Z.; Yu, Z.; Deng, W.; Li, J.; Gao, T.; and Wang, Z. 2022b. Domain Generalization via Shuffled Style Assembly for Face Anti-Spoofing. In *CVPR*.

Wen, D.; Han, H.; Jain, A. K.; et al. 2015. Face spoof detection with image distortion analysis. In *TIFS*.

Xie, J.; Yan, S.; and He, X. 2022. General Incremental Learning with Domain-aware Categorical Representations. In *CVPR*.

Yang, J.; Lei, Z.; Li, S. Z.; et al. 2014. Learn convolutional neural network for face anti-spoofing. In *arXiv*.

Yu, Z.; Li, X.; Niu, X.; Shi, J.; and Zhao, G. 2020a. Face anti-spoofing with human material perception. In *arXiv*.

Yu, Z.; Zhao, C.; Wang, Z.; Qin, Y.; Su, Z.; Li, X.; Zhou, F.; and Zhao, G. 2020b. Searching central difference convolutional networks for face anti-spoofing. In *CVPR*.

Zhang, J.; Tai, Y.; Yao, T.; Meng, J.; Ding, S.; Wang, C.; Li, J.; Huang, F.; and Ji, R. 2021a. Aurora Guard: Reliable Face Anti-Spoofing via Mobile Lighting System. In *arXiv*.

Zhang, K.-Y.; Yao, T.; Zhang, J.; Liu, S.; Yin, B.; Ding, S.; and Li, J. 2021b. Structure Destruction and Content Combination for Face Anti-Spoofing. In *IJCB*.

Zhang, Z.; Yan, J.; Liu, S.; Lei, Z.; Yi, D.; and Li, S. Z. 2012. A face antispoofing database with diverse attacks. In *ICB*.

Zhou, Q.; Zhang, K.-Y.; Yao, T.; Lu, X.; Yi, R.; Ding, S.; and Ma, L. 2023. Instance-Aware Domain Generalization for Face Anti-Spoofing. In *CVPR*.

Zhou, Q.; Zhang, K.-Y.; Yao, T.; Yi, R.; Ding, S.; and Ma, L. 2022a. Adaptive mixture of experts learning for generalizable face anti-spoofing. In *ACM MM*.

Zhou, Q.; Zhang, K.-Y.; Yao, T.; Yi, R.; Sheng, K.; Ding, S.; and Ma, L. 2022b. Generative domain adaptation for face anti-spoofing. In *ECCV*.